(4)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

VLSI PUBLICATIONS

VLSI Memo No. 88-482
October 1988

HIERARCHICAL FLOW CONTROL: A FRAMEWORK FOR SCHEDULING AND PLANNING DISCRETE EVENTS IN MANUFACTURING SYSTEMS

Stanley B. Gershwin

Abstract

This paper discusses the synthesis of operating policies for manufacturing systems. These are feedback laws that respond to potentially disruptive events. We develop laws that are based on realistic dynamic programming models which account for the discrete nature of manufacturing and which are computationally tractable.

These scheduling and planning policies have a hierarchical structure which is systematically based on the production process. The levels of the hierarchy correspond to classes of events that occur with distinct frequencies. At each level, feedback laws select (1) times for the controllable events whose frequency class is treated at that level, and (2) frequency targets for much higher frequency controllable events.

In this hierarchy,

(1) Most calculations deal with expected rates of high frequency activities, conditioned on current states of low frequency activities. There is an important relationship between conditional expected rates at different levels.

(2) Rates are constrained because only one activity (e.g., production operation) can take place at one resource (e.g., machine) at any time.

(3) Rates are found at each level according to a dynamic programming problem for which there exists good approximate solutions. Times for controllable events are chosen to agree with those rates.

Microsystems
Research Center
Room 39-321

Massachusetts
Institute
of Technology

Cambridge
Massachusetts
02139

Telephone
(617) 253-8138

Acknowledgements

To appear in IEEE Proceedings, Special Issue on Discrete Event Systems. This is a revised version of MIT VLSI Memo No. 87-406, July, 1987 . This work was supported in part by the Defense Advanced Research Projects Agency and monitored by ONR under contract N00014-85-K-0213.

Author Information

Gershwin: Department of Mechanical Engineering, Laboratory for Manufacturing and
Productivity, Room 35-331, MIT, Cambridge, MA 02139. (617) 253-2149.

September, 1988 LMP-88-011

HIERARCHICAL FLOW CONTROL: A FRAMEWORK FOR SCHEDULING AND PLANNING DISCRETE EVENTS IN MANUFACTURING SYSTEMS

by

Stanley B. Gershwin

Massachusetts Institute of Technology
Department of Mechanical Engineering
Laboratory for Manufacturing and Productivity
Cambridge, Massachusetts

To be published in the *IEEE Proceedings Special Issue on Discrete Event Systems*.

Revision of LIDS-P-1682

ABSTRACT

This paper discusses the synthesis of operating policies for manufacturing systems. These are feedback laws that respond to potentially disruptive events. We develop laws that are based on realistic dynamic programming models which account for the discrete nature of manufacturing and which are computationally tractable.

These scheduling and planning policies have a hierarchical structure which is systematically based on the the production process. The levels of the hierarchy correspond to classes of events that occur with distinct frequencies. At each level, feedback laws select (1) times for the controllable events whose frequency class is treated at that level, and (2) frequency targets for much higher frequency controllable events.

In this hierarchy,

(1) Most calculations deal with expected rates of high frequency activities, conditioned on current states of low frequency activities. There is an important relationship between conditional expected rates at different levels.

(2) Rates are constrained because only one activity (e.g., production operation) can take place at one resource (e.g., machine) at any time.

(3) Rates are found at each level according to a dynamic programming problem for which there exists good approximate solutions. Times for controllable events are chosen to agree with those rates.

HIERARCHICAL FLOW CONTROL: A FRAMEWORK FOR SCHEDULING AND PLANNING DISCRETE EVENTS IN MANUFACTURING SYSTEMS

Stanley B. Gershwin

Massachusetts Institute of Technology
Cambridge, Massachusetts, USA

1. INTRODUCTION

Operating policies for manufacturing systems must respond to important discrete events such as machine failures, setups, demand changes, expedited batches, etc. These feedback policies must be based on realistic models, and they must be computationally tractable. In this paper, we develop a hierarchical framework for research and algorithm development in scheduling and planning. The structure of the hierarchy is systematically based on the characteristics of the specific kind of production that is being controlled. The levels of the hierarchy correspond to classes of events that have distinct frequencies of occurrence.

Computational tractability is an important concern because of the complexity of the system. Even for a very small, deterministic idealization of a production system, the computational effort for combinatorial optimization renders it impractical for real-time control. Any control scheme must be based on a simplified representation of the system and a heuristic solution of the scheduling problem.

There have been many hierarchical scheduling and planning algorithms, some quite practical and successful. However, there has been no systematic justification of this structure. The main contribution of this paper is a framework for studying and synthesizing such a structure. JES

This work extends a formulation by Kimemia and Gershwin (1983) in which only two kinds of events were considered: production operations on parts and failures and repairs of machines. Operations occurred much more often than failures, and this allowed the use of a continuous representation of material flow. A dynamic programming formulation led naturally to a feedback control policy. The state of the system had two parts: a vector of real numbers represented the surplus, the cumulative difference between production and requirements. The discrete part of the state represented the set of machines that are operational. The object was to choose the production rate vector as a function of the state to keep the surplus near 0.

The production rate (the continuous control variable) was restricted by linear inequality constraints that depended on the repair state. They represented the instantaneous capacity of the system, and they expressed the idea that no machine, while it is operational, may be busy more than 100% of the time; and no machine, while it is not operational, may be used at all. The present paper describes the extension of this work to the widest possible variety of phenomena and decisions in a manufacturing environment.

Figure 1.1 illustrates some of the issues that are considered here. It is a graph of the cumulative production and demand for one part type (j) among many that share one machine (i). A long term production rate (u^1_{ij}) is specified for this part type, and its integral is represented by the solid straight line. It is not possible to follow this line exactly because the machine is set up for Type j parts only during a set of time intervals. During such intervals, the medium term production rate u^2_{ij} must be greater than u^1_{ij}, because during the other intervals -- while it is set up for other parts -- u^2_{ij} is 0. The integral of u^2_{ij} (the dashed line) is staircase-like, close to the integral of u^1_{ij}.

The dashed line cannot be realized either. The machine is unreliable, and while it is down, its production rate u^3_{ij} is 0. Consequently, while it is up and configured for Type j, it must be operated at a short term rate u^3_{ij} greater than that of the dashed line. The dotted line, which represents this phenomenon, is again staircase-like, and is close to the dashed line. Finally, the actual cumulative production graph (which requires too much resolution to be plotted) is a true staircase. It has vertical steps at the instants when parts are loaded, and it is flat otherwise. It is very close to the dotted line.

This paper formalizes this hierarchy, and extends it to an arbitrary number of levels (and distinct classes of events) and several machines.

Literature Survey

There is a large literature in scheduling (Graves, 1981). Many papers are based on combinatorial optimization/integer programming methods (Lageweg, Lenstra, and Rinnooy Kan, 1977 and 1978; Papadimitriou and Kannelakis, 1980) or mixed integer methods (Afentakis, Gavish, and Karmarkar, 1984; Newson, 1975a and 1975b; Wagner and Whitin, 1958). Because of the difficulty of the problem, authors are limited to analyzing computational complexity, or proposing and analyzing heuristics.

An important class of problem formulations is that of hierarchical structure (Bitran, Haas, and Hax, 1981; Dempster et al., 1981; Graves, 1982; Hax and Meal, 1975; and others). The goal is to replace one large problem by a set of many small ones because the latter is invariably easier to treat. These methods are often used but there is no general, systematic way of synthesizing hierarchies for large classes of stochastic scheduling problems. A full critical survey of the hierarchical management and control literature appears in Libosvar (1988a, 1988b).

Multiple time scale problems have recently been studied in the control theory (Saksena, O'Reilly, and Kokotovic, 1984) and Markov chain literature (Delebecque, Quadrat, and Kokotovic, 1984). We use insights from these methods to develop a systematic justification for hierarchical analysis.

This paper also makes use of, and extends the work of Kimemia and Gershwin (1983). The survey in the paper by Maimon and Gershwin (1988) describes this and several related papers. Gershwin (1987a) introduced the version of the hierarchical decomposition that is extended here. That paper did not treat setups (e.g., changes in tooling) in depth. An example which treated setups and failures at two different levels of the hierarchy was described in Gershwin (1987b), and some discussion of setups appeared there. Gershwin, Caramanis, and Murray (1988), describe an improved treatment of this system. Caromicoli (1987) and Caromicoli, Willsky, and Gershwin (1987) treated the same example with methods of the multiple-time-scale Markov literature. Xie (1988) treats a closely related problem -- one with failures only, but two different failure modes, at two different levels -- with the methods of Gershwin (1987a).

Outline

Section 2 describes the manufacturing systems that we are considering. It establishes terminology and discusses the basic concepts for the present approach: capacity and frequency separation. Section 3 builds on the frequency separation to derive a small set of results that form the foundation of the hierarchy. Control in the hierarchy is described in detail in Section 4. Sections 5 and 6 present the two building blocks: the staircase strategy and the hedging point strategy. A simple example appears in Section 7, and conclusions are drawn in Section 8.

2. PRODUCTION EVENTS AND CAPACITY

In this section, we discuss the discrete events that occur during the production process. We define terminology to help describe these events. We categorize events in two ways: the frequency with which they occur; and the degree of control that decision-makers can exert over them. We define capacity, and show how capacity is affected by production events.

2.1 Definitions

A *resource* is any part of the production system that is not consumed or transformed during the production process. Machines -- both material transformation and inspection machines, workers, pallets, and sometimes tools -- if we ignore wear or breakage -- can be modeled as resources. Workpieces and processing chemicals cannot.

An *activity* is a pair of events associated with a resource. The first event corresponds to the start of the activity, and the second is the end of the activity. Only one activity can appear at a resource at any time. For example, drilling 3/8" holes in type 12 parts is an activity: an *operation*. Other examples include machine failures, preventative maintenance, routine calibration, inspection, and training sessions. We use the same term to refer to a set of such pairs of events.

The activities that a resource may be able to perform at a given time may be limited by its current *configuration* or *setup*. These terms refer to physical attributes of a resource -- such as the tooling of a machine -- that can be changed. When to change the configuration (when to *set up*) is an important managerial decision.

Setting up is a special kind of activity. Configurations must be changed so that the full range of material that is demanded can be produced. If it is done too frequently, a resource's capacity for productive work is diminished, but if it is done too infrequently, excessive inventory and response time results. One of the goals of the work presented here is to schedule all activities, including setups.

Examples of production systems that involve setups are machine tools that can support a variety of different cutting edges at different locations, and ion implanters (in semiconductor fabrication) that can support different impurities. In the first case, changing tools involves removing and replacing cutters, and calibration of the new tools. In the second, the chamber must be cleaned. These activities are time-consuming, and must not be done too often.

Let i be a resource and j an activity. Define $\alpha_{ij}(t)$ to be the *activity state* of resource i. This is a binary variable which is 1 if resource i is occupied by activity j at time t, and 0 otherwise. Since at most only one activity may be present at each resource i at a given time,

$$\sum_j \alpha_{ij}(t) \leq 1 \qquad (1)$$

Define $\sigma_{ij}(t)$ to be the *configuration state* of resource i. This is a binary variable which is 1 if resource i is configured for activity j at time t, and 0 otherwise. Assume resource i is set up for activity j at time t. Then if it is not performing j at time t, it will be able to perform j when its current activity ends without losing time or incurring other cost for the changeover.

Configurations are not exclusive the way activities are. Some resources (like flexible machine tools) can be set up for many activities at the same time.

Activity state $\alpha_{ij}(t)$ can be 1 only if $\sigma_{ij}(t)$ is 1. As a consequence, (1) can be made more precise:

$$\sum_{j,\ \sigma_{ij}(t)=1} \alpha_{ij}(t) \le 1 \tag{2}$$

Here, the summation is only over activities j for which resource i is configured at time t, that is, over all activities that can be performed with no change of set up.

Every activity has a *frequency* and a *duration*. To define frequency, let $N_{ij}(T)$ be the total number of times that resource i is occupied by activity j in (0,T). Then define *activity j frequency* (or *rate*) by

$$u_{ij} = \frac{1}{T}N_{ij}(T). \tag{3}$$

This is the frequency with which type j activities occur at resource i. It satisfies $u_{ij} \ge 0$. If activity j is a production operation, then u_{ij} is a production rate. However, u_{ij} may also represent the frequency of doing maintenance, changing setups, experiencing failures, etc.

Let T^{σ}_{ij} be the total time that resource i spends in configurations that allow activity j. Then we can define the *conditional activity j rate*,

$$u^{\sigma}_{ij} = \frac{1}{T^{\sigma}_{ij}}N_{ij}(T). \tag{4}$$

Let τ_{ij} be the average duration of activity j at resource i. It satisfies $\tau_{ij} \ge 0$. Durations may be random or deterministic, but we assume that they are not under the control of the decision-maker.

Observation: If the system is ergodic and in steady state,

$$\tau_{ij}\ u_{ij} = E\alpha_{ij} \tag{5}$$

Proof:

Consider a sample history of the system. The total time that resource i is occupied by activity j in (0,T) is

$$\int_0^T \alpha_{ij}(t)dt. \tag{6}$$

The average duration satisfies

$$\tau_{ij} = \frac{\int_0^T \alpha_{ij}(t)dt}{N_{ij}(T)} = \frac{\frac{1}{T}\int_0^T \alpha_{ij}(t)dt}{u_{ij}} . \tag{7}$$

If the system is ergodic and in steady state, then the time average of a quantity is the same as its expected value, so the numerator is $E\alpha_{ij}$ and (5) is proven. (This can also be viewed as an instance of Little's law.) The assumption that the system is in steady state is an important one. In later sections, the dynamics of the system is divided into subsets, each considered over different time scales. Each subset has a different time period which is required for it to reach steady state.

Since the system is in steady state,

$$\text{prob } (\sigma_{ij} = 1) = \frac{T_{ij}^{\sigma}}{T}. \tag{8}$$

Therefore,

$$u_{ij} = u_{ij}^{\sigma} \text{ prob } (\sigma_{ij} = 1). \tag{9}$$

Note also that

$$E\alpha_{ij} = E(\alpha_{ij} \mid \sigma_{ij} = 1)\text{prob}(\sigma_{ij} = 1) \tag{10}$$

since $E(\alpha_{ij} \mid \sigma_{ij} = 0) = 0$, so that

$$u_{ij}^{\sigma} = \frac{E(\alpha_{ij} \mid \sigma_{ij} = 1)}{\tau_{ij}}. \tag{11}$$

Since only one activity may occur at a resource at one time, the fraction of resource i's time that is spent on activity j is $\tau_{ij}u_{ij}$. This is called the *occupation* of resource i by activity j. We can also define $\tau_{ij}u_{ij}^{\sigma}$ as the *conditional occupation* of i by j, given that it is set up for j. The conditional occupation can be less than 1 if resource i is flexible and is used for other activities in addition to j.

Example: Type 1 parts arrive at Machine 1 at a rate of 1 per hour (u_1). They undergo operations that take 20 minutes (τ_{11}). Therefore Machine 1 is occupied by making Type 1 parts for 1/3 of its time.

2.2 Capacity

From (1),

$$1 \geq E\sum_{j} \alpha_{ij}(t) = \sum_{j} \tau_{ij}u_{ij} \text{ for all resources i.} \tag{12}$$

This is the fundamental capacity limitation: no resource can be occupied more than 100% of the time.

Example: In addition to the Type 1 parts, we wish to send Type 2 parts to Machine 1 for an operation that takes 25 minutes (τ_{12}). There is a demand of one Type 2 part every 35 minutes (u_2). This is not possible because it violates (12).

This can be sharpened, since activities can only occur while the resource is configured for them. By taking the conditional expectation of (1),

$$1 \geq \sum_j E(\alpha_{ij}(t) \mid \sigma_{ij}(t) = 1) = \sum_{j,\ \sigma_{ij}(t)=1} \tau_{ij} u^{\sigma}_{ij} \text{ for all resources } i. \qquad (13)$$

The set of all activity rate matrixes u that satisfies (13) is the *capacity set* Ω. It is important to observe that *capacity is a set* -- a polyhedron -- and not a scalar. Here we have defined capacity as a constant set. In later sections, capacity is described as a function of the state of the system. This means that *capacity is a stochastic set.*

2.3 Frequency Separation

Dynamic models always have two parts: a constant part and a time-varying part. In all dynamic models, there is something that is treated as unchanging over time: some parameters, and, most often, the structure of the model. For example, the model described in Sections 2.1 and 2.2 is a conventional one in which there are static quantities (u_{ij}, u^{σ}_{ij}, τ_{ij}), a static structure, and dynamic quantities ($\alpha_{ij}(t)$, $\sigma_{ij}(t)$, $N_{ij}(t)$).

Recently, the dichotomy between static and dynamic has been extended to systems with multiple time scales, modeled as differential equations or Markov chains. At one end of the scale, there are quantities that are treated as static. The other variables are divided into groups according to the speed of their dynamics. Because of this grouping, it is possible to simplify the computation of the behavior of these systems. Approximate but reasonably accurate techniques have been developed to calculate the effects of the slower and faster dynamics of adjacent groups on each group of variables.

The essential idea is: when dealing with any dynamic quantity, treat quantities that vary much more slowly as static; and model quantities that vary much faster in a way that ignores the details of their variations (such as by replacing fast-moving quantities by their averages or by Brownian noises). This is the central assumption of the hierarchical decomposition presented here.

Assumption 1: The activities can be grouped into sets J_1, J_2, ... such that for each set J_k, there is a characteristic frequency f_k satisfying

$$0 = f_1 \ll f_2 \ll \ldots \ll f_k \ll f_{k+1} \ll \ldots \qquad (14)$$

The activity rates satisfy

$$j \in J_k \Rightarrow f_{k-1} \ll u_{ij} \ll f_{k+1}. \qquad (15)$$

Figure 2.1 represents two kinds of production that satisfy this assumption. The horizontal axis represents frequency and the vertical axis represents occupation of some critical resource. Because of Assumption 1, all the event frequencies occur at distinct clusters.

The time period over which a component of the system reaches steady state depends on the frequency classes of the activities that affect that component. It is on the order of $1/f_{k-1}$ if the lowest frequency activity is a member of J_k.

A capacity set can be associated with each time scale k. Consequently, *capacity is a set of stochastic sets.*

2.4 Slowly varying, piecewise constant rates

In 2.1 and 2.2, u_{ij} is treated as constant. However, it is convenient to allow u_{ij} to be slowly varying. That is, u_{ij} is not constant, but it changes slowly compared to the changes in α_{ij}. An important special case is where u_{ij} is piecewise constant, and its changes occur much less often than those of α_{ij}. Equation (5) is now

$$\tau_{ij}\ u_{ij}(t) = E\alpha_{ij}(t). \tag{16}$$

This is established in the same manner as (5), but the bounds of the integral (6) are t_1 and t, where t_1 is the time of the most recent change in u_{ij}, and t is the current time. The quantity $u_{ij}(t)$ satisfies

$$N_{ij}(t) = \int_0^t u_{ij}(s)ds \text{ for } \tau_{ij} > 0, \text{ or}$$

$$N_{ij}(t) - N_{ij}(t_1) = \int_{t_1}^t u_{ij}(s)ds = (t - t_1)u_{ij}(t_1). \tag{17}$$

The assumption here is that many occupations of resource i by activity j occur in the interval (t_1, t): enough so that

$$E\alpha_{ij}(t) = \frac{1}{t-t_1}\int_{t_1}^t \alpha_{ij}(s)ds. \tag{18}$$

2.5 Degree of controllability and predictability

Events may or may not be under the control of the decision-maker. For the purpose of this paper, we say that an event is *controllable* if its time of occurrence may be chosen, whether or not there are constraints on that choice. An event is uncontrollable otherwise. An activity is controllable if its initial event is controllable, and uncontrollable otherwise. Operations are controllable, for example, and failures are not.

Gershwin (1987a) discusses differing degrees of controllability among activities in a manufacturing system: for example, failures are completely uncontrollable. On the other hand, the manager has almost complete freedom in choosing the time of the next operation or setup change activity, but the activity must be one of a limited set of operations or setups. In addition, lunch breaks and holidays can be thought of as perfectly predictable, but completely uncontrollable activities.

We do not distinguish, in the notation developed here, between controllable, uncontrollable, predictable, and unpredictable events. However, the specific formulations of the dynamic programming problems (the hedging point problems) described below must take this into account.

2.6 Effects of events

The goal of the factory is to produce in a way that satisfies demand at least cost. The only events that directly further this goal are the production events (operations), and only if they are chosen correctly. The direct effects of all the other events (failures, setups, maintenance, etc.) work against this goal.

When any activity occurs, it prevents all other activities from occurring at the same resource. Thus a low frequency, high occupation activity is a major disruption to the system. During such an activity, the resource it occupies is unavailable for a very long time (as seen by the high frequency events). This may not simply shut down all production; instead, it may temporarily restrict only some kinds of production. Such disruptions greatly complicate the scheduling problem.

2.7 Purpose of the decomposition

It is possible to represent the scheduling problem as an integer program, particularly if time is discretized. However, this almost always leads to a problem which cannot practically be solved even in the absence of random events. Heuristics are often employed, and are sometimes useful for specific cases. The goal of the approach described below is to formulate the problem in a way that will provide a general methodology for developing approximate feedback solutions for stochastic scheduling problems in manufacturing systems.

The solution approach is based on a reformulation of the problem in which the large set of binary variables that indicate the precise times of events is replaced by a small set of real (continuous) variables representing the rates at which events occur. This is a good approximation because of the large differences in frequencies among these events. Eventually, the binary variables are calculated, but by a much simplified procedure.

3. THE SPECTRUM AND THE HIERARCHY

In this section, we define the variables of the hierarchy and we describe the purposes of the calculations that take place at each of the levels. In the following sections, we propose problem formulations for those calculations. In Sections 3.1 and 3.2, we treat a restricted version of the hierarchy: one without configuration changes. Setups are introduced into the hierarchy in Sections 3.3 and 3.4.

3.1 Definitions

3.1.1 Levels and frequencies

The structure of the hierarchy is based on Assumption 1: that events tend to occur on a discrete spectrum. Classes of events have frequencies that cluster near discrete points on the spectrum. The control hierarchy is tied to the spectrum. Each level k in the hierarchy corresponds to a discrete point on the spectrum and thus to a set of activities. This point is the characteristic frequency f_k (and $1/f_k$ is the characteristic time scale) of those activities.

At each level of the hierarchy, events that correspond to higher levels (i.e., lower frequencies, and smaller values of k) can be treated as discrete and constant or slowly varying. Events that correspond to lower levels can be described by continuous (real) variables. These variables can be treated as though they are deterministic.

The approach is to define a set of rate or frequency variables for every activity. These quantities represent the behavior of the system in an aggregated way. At each level, we calculate optimal values for those aggregate variables. Optimal, here, means that they must be close, on the average, to the corresponding values chosen at the higher levels. However, they must respond to events that occur at their own level.

Define the *level* L(j) of activity j to be the value of k in Assumption 1 associated with this activity. That is,

$$L(j) = k \text{ if } j \in J_k \tag{19}$$

in (15). We choose the convention that less frequent activities are higher level activities and have smaller values of k; lower levels have larger values of k.

We introduce the concept of a level k observer or manager. Such an observer has a precise model only of events that occur with frequencies near f_k (in the sense of Assumption 1). In some cases, there is such a model because the observer/manager affects the events. The observer has greatly simplified models of events that occur at frequencies far from f_k.

Quantities that change with much lower frequencies (such as production goals that are selected by higher level managers) are treated as constant. Even after they change, the observer is not able to anticipate future changes. Events that occur with much higher frequencies -- such as changes in α or σ -- cannot be distinguished in detail. This observer also cannot see activities whose duration is much less than $1/f_k$.

The level k observer is able to see frequencies of lower level (higher frequency) events, and may even have a model of how the frequencies change. However, changes in the frequencies are themselves events. For a level k observer to have a precise model of changes in higher frequencies, the changes must occur at frequencies near f_k. The hedging point problem is a method of selecting frequencies of high frequency events. It is described below.

In 3.1.3, we define level k quantities. These are values of system states as perceived by a level k observer. The frequencies of high frequency events, as seen by this observer, depend on the current states of low frequency activities, and expectations must be conditioned on the current states of low frequency activities. First we present an example to make this concrete.

3.1.2 Example

Consider a system in which there are two machines that work in parallel performing operations on a single part type. Each machine is maintained four times per year (and is down a full week each time); failures happen roughly once per week (and occupy 5% of each machine's time while it is not undergoing maintenance); and operations take one hour. The system is run 24 hours per day.

A hierarchy can be constructed with long range planning at level 1, maintenance at level 2, failures at level 3, and operations at level 4. The level 1 manager chooses long range average maintenance frequencies and operation rates (the former in accordance with the instructions of the manufacturer of the machines or according to the company's own studies on the relationship between preventative maintenance and reliability; the latter to satisfy long range demand forecasts). This manufacturing system is capable of producing 1.754 parts per hour, in the long run (because one machine is available 8 weeks out of 52, and two machines are available 44 weeks out of 52, and each machine works 95% of the time it is available).

The activity states are α_{ij}, where i = machine number = 1 or 2; and j = activity number. Maintenance is j = 1; j = 2 is failure; j = 3 is production operation. Thus, the level 1 capacity set is

$$0 \le u^1_{13} \le .877$$
$$0 \le u^1_{23} \le .877$$

Assume that the long range demand rate is selected to be $u^1_{13} + u^1_{23} = 1.6$ parts per hour.

The level 2 manager selects and announces the actual times to perform maintenance, and calculates the production rates $u^2_{i3}(\alpha_{11},\alpha_{21})$ during the periods that the system is in each maintenance state. The capacity set is

$$0 \le u^2_{13}(0,0) \le .95$$
$$0 \le u^2_{13}(0,1) \le .95$$
$$0 = u^2_{13}(1,0)$$
$$0 = u^2_{13}(1,1)$$

$0 \leq u_{23}^{2}(0,0) \leq .95$
$0 \leq u_{23}^{2}(1,0) \leq .95$
$0 = u_{23}^{2}(0,1)$
$0 = u_{23}^{2}(1,1)$

A possible set of production rates is: the system will produce $u_{13}^{2}(0,1) = u_{23}^{2}(1,0) = .9$ parts per hour during periods while one machine is undergoing maintenance, and $u_{i3}^{2}(0,0) = .864$ parts per hour (i = 1,2) during periods while no machine is undergoing maintenance. The average production rates are $u_{13}^{1} = u_{23}^{1} = .8$, since both machines are available for 44/52 of the time, machine 1 is available and machine 2 is being maintained for 4/52 of the time, and machine 2 is available and machine 1 is being maintained for 4/52 of the time

Note that this manager is only concerned with the average effects of failures, but is concerned with the actual maintenance states.

The level 3 manager must choose production rates $u_{i3}^{3}(\alpha_{11},\alpha_{21},\alpha_{12},\alpha_{22})$ for each maintenance state and each failure state. The targets are $u_{i3}^{2}(\alpha_{11},\alpha_{21})$. Now the capacity sets are (i = 1,2):

$0 = u_{i3}^{3}(0,1,1,0)$ (machine 2 being maintained; machine 1 under repair)
$0 = u_{i3}^{3}(1,0,0,1)$ (machine 1 being maintained; machine 2 under repair)
$0 \leq u_{13}^{3}(0,1,0,0) \leq 1$ (machine 2 being maintained; neither machine under repair)
$0 = u_{23}^{3}(0,1,0,0)$ (machine 2 being maintained; neither machine under repair)
$0 = u_{13}^{3}(1,0,0,0)$ (machine 1 being maintained; neither machine under repair)
$0 \leq u_{23}^{3}(1,0,0,0) \leq 1$ (machine 1 being maintained; neither machine under repair)
$0 = u_{i3}^{3}(0,0,1,1)$ (neither machine being maintained; both machines under repair)
$0 \leq u_{13}^{3}(0,0,0,1) \leq 1$ (neither machine being maintained; machine 2 under repair)
$0 = u_{23}^{3}(0,0,0,1)$ (neither machine being maintained; machine 2 under repair)
$0 = u_{13}^{3}(0,0,1,0)$ (neither machine being maintained; machine 1 under repair)
$0 \leq u_{23}^{3}(0,0,1,0) \leq 1$ (neither machine being maintained; machine 1 under repair)
$0 \leq u_{i3}^{3}(0,0,0,0) \leq 1$ (neither machine being maintained or under repair)

A set of feasible rates is (i = 1,2):

$u_{i3}^{3}(0,1,1,0) = 0$
$u_{i3}^{3}(1,0,0,1) = 0$
$u_{13}^{3}(0,1,0,0) = .947$
$u_{23}^{3}(0,1,0,0) = 0$
$u_{13}^{3}(1,0,0,0) = 0$
$u_{23}^{3}(1,0,0,0) = .947$

$u_{i3}^{3}(0,0,1,1) = 0$
$u_{13}^{3}(0,0,0,1) = .909$
$u_{23}^{3}(0,0,0,1) = 0$
$u_{13}^{3}(0,0,1,0) = 0$
$u_{23}^{3}(0,0,1,0) = .909$
$u_{i3}^{3}(0,0,0,0) = .909$

The average of the level 3 production rates of all the states in which $(\alpha_{13},\alpha_{23}) = (0,1)$ are $u_{13}^{2}(0,1) = .9$ (since machine 1 is down with probability .05 and up with probability .95), and $u^2_{23}(0,1) = .0$. Similarly, the average production rates of the states in which $(\alpha_{13},\alpha_{23}) = (1,0)$ are $u^2_{23}(1,0) = .9$ and $u^2_{13}(1,0) = .0$. The other averages of level 3 production rates are the corresponding level 2 quantities.

The level 3 quantities are chosen so that their expectations are the corresponding level 2 quantities, and the level 2 quantities are chosen so that their expectations are the corresponding level 1 quantities. These are conditional expectation relationships, and they are generalized below. A more sophisticated choice of lower level rates that satisfy the upper level expectations is called the hedging point policy, and is described below. This policy allows the rates to depend on the amount of material produced, as well as the states of the machines.

The task of the level 4 manager (or worker, or material handling system) is to load parts onto the machines (i.e., to choose α_{i3}) to meet the last set of rates. The staircase policy, which is described below, is one possible way of doing this. Whenever, the number of type j parts loaded on machine i is less that the time integral of u^3_{ij}, a new part is loaded.

3.1.3 Level k states and rates

In this section, we develop the capacity sets at each level of a hierarchy, and the relationships among rates at each level. In later sections, we will describe how to calculate $u^k(\alpha,\sigma)$ (the hedging point strategy) and how to determine times for discrete events (such as maintenance at level 2 and loading of parts at level 4).

Let $\alpha^k_{ij}(t)$ be the *level k activity state of resource i.* This is defined, only for activities j whose level is k or higher, as

$$\alpha^k_{ij}(t) = \alpha_{ij}(t) \text{ for } L(j) \le k. \qquad (20)$$

Define α^k as the matrix whose components are α^k_{ij}. Its dimensionality depends on k. It is the low frequency part of α, whose components change at frequencies much less than, or roughly equal to f_k. It is the only part of α accessible to a level k observer.

In the example,

$$\alpha^2 = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \end{bmatrix}$$

$$\alpha^3 = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}$$

Let E_k be the *level k expectation operator.* It is the conditional expectation, given that all level m quantities ($\alpha^m_{ij}(t)$, $m \le k$) remain constant at their values at time t. That is, for any random variable z,

$$E_k z(t) = E(z \mid \alpha^m_{ij} = \alpha^m_{ij}(t), \text{ for all i, j, such that } L(j) \le m \le k). \qquad (21)$$

Let u^k_{ij} be the *level k rate of activity j* at resource i. It is defined only if the level of activity j is lower than k, that is, $L(j) > k$. This is a representation of the high frequency part of α that makes sense to the level k observer. The level k rate of activity j is the frequency that a level k observer would measure that activity j occurs while all level m events ($m \leq k$) are held constant at their current values. This rate is defined as

$$u_{ij}^{k}(t) = \frac{E_k \alpha_{ij}(t)}{\tau_{ij}} \text{ for } L(j) > k. \tag{22}$$

Note that

$$u_{ij}^{k}(t) \geq 0. \tag{23}$$

It is important to note that u^k is a function of the current value of α^k, the level k activity state. Thus, u^k is a stochastic process. In the example, u^1 is a constant, u^2 is a function of α^2, and u^3 is a function of α^3.

The conditioning event of E_k is a subset of that of E_{k-1}. This is because the set of quantities held constant for E_{k-1} is a subset of that for E_k. Consequently,

$$E_{k-1} E_k \alpha_{ij} = E_{k-1} \alpha_{ij}. \tag{24}$$

Taking the level k-1 expectation of (22):

$$E_{k-1} u_{ij}^{k} = E_{k-1} \frac{E_k \alpha_{ij}(t)}{\tau_{ij}}. \tag{25}$$

But this is equal to $\frac{E_{k-1} \alpha_{ij}(t)}{\tau_{ij}}$ according to (24). This implies that

$$E_{k-1} u_{ij}^{k} = u_{ij}^{k-1}. \tag{26}$$

That is, the level k-1 rate of an activity is the level k-1 expectation of the level k rate of the activity. This is a very important observation, because it relates quantities at different levels of the hierarchy.

In the example, (26) is satisfied by all rates. In particular,

$$E_1 u_{13}^{2} = \frac{4}{52} u_{13}^{2}(0,1) + \frac{4}{52} u_{13}^{2}(1,0) + \frac{44}{52} u_{13}^{2}(0,0) = .8 = u_{13}^{1}$$
$$E_2 u_{13}^{3}(t_1) = (.05) u_{13}^{3}(0,1,1,0) + (.95) u_{13}^{3}(0,1,0,0) = .9 = u_{13}^{2}(0,1)$$

where t_1 is any time that $\alpha_{11} = 0$ and $\alpha_{21} = 1$.

Recall that the rate matrix u_{ij}^{k} is defined only for $L(j) > k$. Therefore the dimensionality of u^k is generally smaller than that of u^{k-1}. Another way of writing (26) is

$$E_{k-1} u^{k} = \text{proj}\,(u^{k-1}, k), \tag{27}$$

where proj (z,k) is the projection of vector z onto the space of u^k.

If L(j) > k, level k of the hierarchy calculates u_{ij}^k. How that calculation is performed depends on the degree of control of activity j. If activity j can be initiated by the decision-maker rather than by nature, then u_{ij}^k is chosen to satisfy (26).

All activities j appear in three different guises in the hierarchy. At their own level (k=L(j)), they appear as pairs of discrete events (the start and the end of the activity). This is, of course, exactly what they are. No approximate representation is possible. At higher levels in the hierarchy (k<L(j)), however, their details are ignored, and they are represented by rates (u_{ij}^k). At lower levels (k>L(j)), they are treated as constant at their current values.

Controllable activities are chosen from top down. That is, a rate u_{ij}^1 is chosen initially. Then (k>1) is chosen to satisfy (26) and other conditions (according to the *hedging point strategy* of Section 6) for increasing values of k until k = L(j). At that point, α_{ij} is chosen to satisfy (22) according to the *staircase strategy* described in Section 5.

On the other hand, (22) and (26) have different interpretations when activity j is not controllable (for example, machine failures). In that case, the expectations are statistical operations, in which data are collected and sample means are found. The rate $u_j^{L(j)-1}$ is calculated from (22) by observing the values of α_{ij}. If L(j)<k-1, (26) is repeated for decreasing values of k.

3.2 Capacity in the hierarchy

For each k, the sum in (1) can be broken into two parts:

$$\sum_{j,\,L(j)>k} \alpha_{ij} \le 1 - \sum_{j,\,L(j)\le k} \alpha_{ij}^k \tag{28}$$

in which (20) is applied to the high-level sum on the right side. If we take a level k expectation of (28), the right side is not affected. From (22),

$$\sum_{j,\,L(j)>k} \tau_{ij} u_{ij}^k \le 1 - \sum_{j,\,L(j)\le k} \alpha_{ij}^k \tag{29}$$

This equation is the basic statement of capacity in the hierarchy. It limits the rates at which lower level events can occur as a function of the current states of higher level events. If any high level activity is currently at resource i, that resource is not available for any low level events. In that case, the right side of (29) is 0 and all u_{ij}^k that have a positive coefficient must be zero. If none of the higher level activities in (29) are currently taking place, this inequality becomes

$$\sum_{j,\,L(j)>k} \tau_{ij} u_{ij}^k \le 1. \tag{30}$$

In the example, the feasible set of values for u^k was exhibited as a function of α^k .

Capacity is thus a function of hierarchy level and, since it depends on the state of the system, a stochastic function of time. We define the *level k capacity set* as

$$\Omega^k(\alpha^k) = \left\{ u^k \;\middle|\; \sum_{j,\, L(j)>k} \tau_{ij} u^k_{ij} \le 1 - \sum_{j,\, L(j)\le k} \alpha^k_{ij} \quad \forall\, i; \qquad u^k_{ij} \ge 0 \quad \forall\, j,\, L(j)>k \right\}. \tag{31}$$

This set is the constraint on the hedging point strategy (Section 6). It limits the choice of rates u^k as a function of the current state of the system. Note that the condition

$$u^k \in \Omega^k(\alpha^k) \tag{32}$$

is a necessary but not sufficient condition. That is, $\Omega^k(\alpha^k)$ was constructed so that every sequence of events must satisfy (32). However, we have not demonstrated that for every u^k that satisfies (32) there corresponds a feasible sequence of events. We assume sufficiency in the following, however.

The configuration state is added to the definition of the capacity set in Section 3.4.

3.3 Setups in the hierarchy

Setups, or configuration changes, are special activities. They are distinctive in that after a setup is performed, the set of activities that can occur is changed. Note also that σ can be changed only after some specific activity (a setup change) is performed.

To incorporate setups in the hierarchy, we make the following observations:

1. Changes of setup are activities that share many characteristics with other activities, such as operations. Thus, while the setup of a machine is being changed, the machine cannot be occupied by any other activity (that is, it cannot be used to perform operations, it cannot undergo maintenance, and it is fair to assume that it cannot fail.) Symbolically, changing resource i from being able to perform activity j_1 to being able to perform activity j_2 is itself an activity j_3; and j_1, j_2, and j_3 are among the values of j that appear in (1).

2. Since changes of setups are activities, they have frequencies. Therefore, setup changes appear at appropriate levels in the hierarchy. If there were only one kind of setup change, the system would look very different above the setup change level than below.

(a) Above this level, changes of setup could not be observed because they occur too frequently, and they take too little time to do. The observer would be aware that there is an activity called "changing setups," but it would only know its frequency. Since changes in setup occur at great speed, the system would appear always to be set up for all activities.

(b) Below the level of setup changes, the system would appear to be always, or never, set up for each activity.

3. More generally, there may be many different setup changes, and they may appear at different hierarchy levels. For example, there may be major and minor setups, in which changes among broad classes of parts are done infrequently, and changes within the classes occur frequently.

Assume L(j) > k. Let σ^k_{ij} be the *level k configuration state.* It is given by

$$\sigma^k_{ij}(t) = \begin{cases} 1 \text{ if activity j can be performed at resource i before time } t+\delta, \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{33}$$

where $\delta << \frac{1}{f_k}$.

Observation: If $\sigma^k_{ij}(t) = 1$, then $\sigma^{k-1}_{ij}(t) = 1$.

Proof:

Since $\sigma^k_{ij} = 1$, activity j can be initiated at resource i before time t+δ, where $\delta <<$ $1/f_k$. But $1/f_k << 1/f_{k-1}$. Therefore, activity j can be initiated at resource i before time t+δ, where $\delta << 1/f_{k-1}$, so $\sigma^{k-1}_{ij} = 1$.

Similarly, if $\sigma_{ij}(t) = 1$, then $\sigma^k_{ij}(t) = 1$. Recall that α and σ without superscripts refer to the true activity and configuration states.

We must extend the definition of level k expectation to include σ:

$$E_k z(t) = E(z \mid \alpha^m_{ij} = \alpha^m_{ij}(t),\ \sigma^m_{ij} = \sigma^m_{ij}(t), \text{ for all i, j, such that } L(j) \le m \le k). \tag{34}$$

Using this definition of E_k, we still use (22) to define u^k. Thus, u_k is now a function of the current level k setup (σ^k) as well as the current level k activity state (α^k).

Define the *level k conditional activity rate,*

$$u^{\sigma k}_{ij} = \frac{E_k(\alpha_{ij} \mid \sigma^k_{ij} = 1)}{\tau_{ij}} \text{ if } L(j) > k. \tag{35}$$

This is the frequency at which resource i performs type j activities during the time that it is configured for that activity at level k.

By the same reasoning that led to (9),

$$u^k_{ij} = u^{\sigma k}_{ij}\ prob(\sigma^k_{ij} = 1) \tag{36}$$

3.4 Capacity and setups in the hierarchy

If L(j) > k, then

$\sigma_{ij}^k = 0 \Rightarrow \alpha_{ij}^k = 0$ and $u_{ij}^k = 0$.

Therefore, if we take a level k expectation of (28), now according to (34), we find

$$\sum_{\substack{j,\ \sigma_{ij}^k=1,\\ L(j)>k}} \tau_{ij} u_{ij}^k \le 1 - \sum_{j,\ L(j)\le k} \alpha_{ij}^k \tag{37}$$

We now define the level k capacity set as

$$\Omega^k(\alpha^k,\sigma^k) = \left\{ u^k \;\middle|\; \sum_{\substack{j,\ \sigma_{ij}^k=1,\\ L(j)>k}} \tau_{ij} u_{ij}^k \le 1 - \sum_{j,\ L(j)\le k} \alpha_{ij}^k \;\forall\; i; \quad u_{ij}^k \ge 0; \right.$$

$$\left. u_{ij}^k = 0 \text{ if } \sigma_{ij}^k = 0 \quad \forall\; i,\ j,\ L(j)>k \right\}. \tag{38}$$

4. CONTROL IN THE HIERARCHY

The goal of the hierarchical scheduler is to select a time for each controllable event. This is performed by solving one or two problems at each level k. We emphasize control -- i.e., scheduling and planning -- here. Data-gathering and processing is also an important function of the hierarchy, but is not discussed in this paper. The hierarchy is illustrated in Figure 4.1.

Problem 1: (The hedging point strategy)

Find u_{ij}^k (for all j, L(j)>k) satisfying (26) and (37) (and possibly other conditions).

Problem 2: (The staircase strategy)

Find α_{ij}^k (for all j, L(j)=k) satisfying

$$E_{k-1}\alpha_{ij}^k = \tau_{ij}u_{ij}^{k-1} \tag{39}$$

(and possibly other conditions).

At the top level of the hierarchy (k=1), required rates of some of the controllable activities are specified as input data, for example, production rates and maintenance frequencies. Other rates may not be specified, such as setup frequencies. We assume that rates of uncontrollable events are known. The frequency associated with the top level is 0. Consequently, there is no Problem 2 at that level, and Problem 1 reduces to a static optimization to determine the unspecified rates of controllable activities. The function of Problem 1 here is to choose all the rates that are not specified. The vector u^1 is the target rate vector for level 2.

Recall that at each the level of the hierarchy, there exists some activity that can be described as a time-varying set of discrete events. These activities may or may not be controllable. At level k > 1, if there are any controllable activities, we solve Problem 2. (An example is the change in setup of a machine.) Controllable activities are thereby initiated in such a way that their rates of occurrence are close to the target rates that are determined at level k-1.

Then we solve Problem 1 to determine the level k rates of occurrence u_{ij}^k of all activities j whose frequencies are much higher than f_k. These rates are refinements of the target rates determined at level k-1: u_{ij}^{k-1}. They differ from the higher level rates in that they are affected by the level k discrete events. These events, if they are controllable, were chosen by Problem 2 at this level. However, even if the level k events are not controllable, the level k rates differ from the higher level rates. These rates are then the targets for level k+1.

For example, if at level k we choose setup times, the production rates must be calculated so that they are appropriate for the current setup. If we are making Type 1 parts at the rate of 4 per day, but the necessary machine is only set up for that part on Tuesdays, then we must work at a rate of 20 per day on Tuesday and 0 Type 1 parts per day during the rest of the week.

Similarly, the activities associated with level k may not be controllable, such as machine failures. It is still necessary to refine the production rates. If the overall requirements for Type 1 parts are 20 per day, and the machine is down 10% of the time, *and failures occur several times per day*, then the appropriate strategy is to operate the machine at a rate of 22.2 parts per day while it is up. Note that this only makes sense if failures are much more frequent than setups and much less frequent than operations. If not, related but different calculations must be performed in a different order. That is, a different hierarchy is appropriate.

An important feature of this hierarchy is that rates u_{ij}^k are always chosen to be within the current capacity of the system. When a level m event occurs ($m \leq k$), the capacity set (38) changes. Problem 2 is then re-solved to keep the rates feasible. As mentioned earlier, this is necessary for feasibility. In all the simulation experiments that we have performed, it appeared to be sufficient as well.

5. THE STAIRCASE STRATEGY -- PROBLEM 2

The staircase strategy was introduced by Gershwin, Akella, and Choong (1985) and Akella, Choong, and Gershwin (1984), although stated somewhat differently. It was used to load parts in a simulation of a flexible manufacturing system.

Instead of treating the statement of Problem 2 in Section 4 directly, we choose starting times for events α^k_{ij} to satisfy (3), or rather

$$N^k_{ij}(t) \sim \int_0^t u^k_{ij}(s)ds \tag{40}$$

where $N^k_{ij}(t)$ is the number of times activity j occurs at resource i during [0,t]. This expression is only approximate because the left side is an integer and the right side is a real number. The objective is to develop an algorithm which keeps the error in (40) less than 1. This is because, approximately,

$$E_{k-1}\alpha^k_{ij}(t) = \frac{1}{t-t_1}\int_{t_1}^t \alpha_{ij}(s)ds = \frac{1}{t-t_1}\Big(N^k_{ij}(t)-N^k_{ij}(t_1) \Big)\tau_{ij} \tag{41}$$

in steady state. If the times to start activities are chosen to satisfy (40), then

$$E_{k-1}\alpha^k_{ij}(t) = \tau_{ij}u^k_{ij}(t_1) \tag{42}$$

The difference between (40) and (39) is that a simple algorithm can be devised to implement (40). It is called the *staircase strategy* because of the graph of $N^k_{ij}(T)$.

Staircase strategy: For all activities j such that L(j) = k, perform activity j at resource i as early as possible after the *eligibility rule* is satisfied.

$$\textit{Eligibility rule: } N^k_{ij}(T) \le \int_0^T u^k_{ij}dt \tag{43}$$

If there were only one activity in the system, it would be initiated as soon as (43) were satisfied with equality. Immediately afterward, the left side of (40) would exceed the right side by exactly 1. The difference would then start to diminish until, again, (43) is satisfied with equality. Thus, the error in (40) would never grow larger than 1. Figure 5.1 represents this strategy, and illustrates the term "staircase." The solid line represents the right side of (43), and the dashed line represents the left side. Note that the change in slope of the solid line poses no difficulties for this strategy.

Example: If activity j is an operation on Type A parts at Machine 6, attempt to load a Type A part into the machine whenever (43) is satisfied.

In reality, there are two complications. First, because there are other activities, activity j may not be the only one to satisfy (43) at any instant. Therefore, there must be a mechanism or an additional eligibility rule for selecting one. Consequently, we can no longer assert that (40) is satisfied with an error no larger than 1.

Second, there are relationships among activities other than non-simultaneity. For example, some manufacturing operations may not be performed unless the system is set up in a certain way. That is, in order to perform an operation on Type 1 parts, the system must be set up for them. The most recent setup activity must have been one that is appropriate for Type 1 parts. This leads to additional eligibility rules.

Example: If activity j is an operation on Type A parts at Machine 6, attempt to load a Type A part into the machine whenever (43) is satisfied and Machine 6 is set up for Type A and the part that has been waiting longest for Machine 6 that can be produced in its current configuration is Type A.

Methods for implementing this strategy can be developed based on the methods of Ramadge and Wonham (1985), Maimon and Tadmor (1986).

6. THE HEDGING POINT STRATEGY -- PROBLEM 1

The hedging point strategy was introduced by Kimemia and Gershwin (1983) and refined by Gershwin et al. (1985) for a restricted version of the scheduling problem discussed here. In that problem, there were only two activities: operations and failures. The hedging point strategy was used to calculate the production rates of parts in response to repairs and failures of machines.

In the present context, the purpose of the problem is to find u^k_{ij} (for all j such that $L(j) > k$) to satisfy (26) and (32) (and possibly other conditions). That is, we find the optimum frequencies of controllable events whose frequencies are much higher than f_k. These frequencies are chosen in response to changes in low frequency activities: those whose values change at a frequency roughly f_k or slower.

6.1 Surplus

We introduce x^k_{ij}, the *activity j surplus*. This quantity represents the excess of occurrences of activity j as determined by u^k_{ij} over the number of occurrences required by u^{k-1}_{ij}. The surplus is illustrated in Figure 6.1. It satisfies

$$x^k_{ij}(t) = \int_0^t u^k_{ij}(s)ds - \int_0^t u^{k-1}_{ij}(s)ds \tag{44}$$

or

$$\frac{dx^k_{ij}}{dt} = u^k_{ij} - u^{k-1}_{ij}. \tag{45}$$

To satisfy (26), i.e., to keep u^k_{ij} near u^{k-1}_{ij}, we must keep x^k_{ij} near 0. We therefore define a strictly convex function g such that $g(0) = 0$; $g(x) \geq 0 \ \forall \ x$; and $\lim_{||x||\to\infty} g(x) = \infty$; and we seek u^k_{ij} to minimize

$$E_{k-1}\int_0^T \sum_{ij} g(x^k_{ij}(t))dt \tag{46}$$

in which T is long enough so that the dynamic programming problem has a time-invariant solution $u^k_{ij}(x^k,\alpha^k)$. Thus T is much greater than $1/f_k$. If (46) is small, then $x^k_{ij}(t)$ must be small for all t. Equation (44) then implies that $u^k_{ij}(t)$ is near u^{k-1}_{ij}.

6.2 Capacity constraints

The activity rate vector $u^k(t)$ must satisfy the stochastic capacity constraints

$$u^k(t) \in \Omega^k(\alpha^k(t),\sigma^k(t)) \tag{47}$$

where $\Omega^k(\alpha^k(t),\sigma^k(t))$ is given by (38). This means that the activity rates of all high frequency activities are restricted in a way that depends on the current states of activities whose frequencies are roughly f_k or less. Those whose frequencies are much less than f_k can be treated as constant at their present values, but the variations of those that change at a frequency comparable to f_k must be considered.

Because Kimemia and Gershwin were dealing with machine failures and repairs, they could treat $\alpha^k(t)$ as the state of a Markov process. Here, however, some components of $\alpha^k(t)$ are chosen by the scheduler according to the staircase strategy of Section 5. For the purpose of determining the frequencies of high-frequency activities, we treat $\alpha^k(t)$ as though it is generated by some exogenous stochastic process with transition rate matrix λ^k:

$$\lambda^k_{\alpha\beta}\delta t = \text{prob}\,\{\alpha^k(t+\delta t)=\beta \mid \alpha^k(t)=\alpha\},\ \alpha\neq\beta\ ; \qquad \lambda^k_{\alpha\alpha} = -\sum_{\beta\neq\alpha}\lambda^k_{\alpha\beta}. \tag{48}$$

By treating all level k events this way, we are ignoring information that could be used, in principle, to improve the performance function (46). Since the time for the next event is known and not random, the optimal trajectory should be different. This requires further study.

Note that σ^k changes only after some changes in α^k. That is, in order for the setup of a machine to change, a setup-change activity must be completed. Therefore, we need only consider the dynamics of α^k and the relationship between σ^k and α^k.

6.3 Other constraints

6.3.1 Specified rates

Some activities are non-controllable, such as machine failures. Their frequencies cannot be chosen; they are given quantities. Thus, if $\mathcal{N}$ is the set of uncontrollable activities,

$$u^k_{ij} \text{ specified, } j \in \mathcal{N}. \tag{49}$$

6.3.2 Conservation of flow

Let resource 1 be machine 1 and resource 2 be machine 2. Let activity j on resources 1 and 2 be operations on type j parts. If type j parts go only to machine 1 and then machine 2, then

$$u^k_{1j}(t) = u^k_{2j}(t). \tag{50}$$

In general, relationships like (50) hold whenever the amount of time that parts spend in the system is small compared with $1/f_k$. If the time is not small, then events can occur with non-negligible probability at machine 2 after parts enter machine 1 that cause (50) to be violated. This limitation is important. Van Ryzin (1987) and Lou, Van Ryzin, and Gershwin (1987) study alternatives to (50) when delay in the system is significant.

6.3.3 Setups

Other activities require special constraints because of their special nature. For example, when a resource may have more than one configuration, and setups require significant time, setup frequencies are constrained to satisfy a set of equality constraints. Assume resource i has configurations 1, ..., C(i). Denote $j=(iab)$ as the activity of changing the configuration of resource i from a to b. Then u^k_{iab} is the level k frequency of changing the configuration of resource i from a to b. These frequencies must satisfy

$$\sum_a u^k_{iab} = \sum_a u^k_{iba}. \tag{51}$$

since the frequency of changing into setup b must be the same as the frequency of switching out of setup b. Related formulations appear in Gershwin (1986) and Choong (1988).

6.3.4 Summary

We summarize all such miscellaneous constraints as

$$m(u^k(t)) = 0. \tag{52}$$

6.4 Problem statement

Here we present a compact statement of the problem. It is a dynamic programming problem whose states are $x^k(t)$, $\alpha^k(t)$, and $\sigma^k(t)$ and whose control is $u^k(t)$. (The rates u^{k-1} are treated as exogenous constants.)

Find the feedback control law $u^k(x^k(t),\alpha^k(t),\sigma^k(t),t)$ to minimize (46) subject to (45), (47), and (52) in which α^k is the state of an exogenous Markov process, with parameters λ^k. The initial conditions at t=0 are $x^k(0)$, $\alpha^k(0)$ $\sigma^k(0)$. T is very large.

6.5 Solution

Kimemia and Gershwin (1983) derived a Bellman's equation for this problem:

$$0 = \min \left\{ g(x^k) + \frac{\partial J}{\partial x}\left(u^k - \text{proj }(u^{k-1},k)\right) + \frac{\partial J}{\partial t} + \sum_\beta \lambda_{\alpha\beta} J[x^k,\beta,\sigma(\beta),t] \right\} \tag{53}$$

in which $J[x^k(t),\alpha^k(t),x^k(t),t]$ is the cost-to-go function, the cost incurred during (t,T) if the initial conditions are $\alpha^k(t)$ and $\sigma^k(t)$ at time t; $\sigma(\beta)$ is the new configuration state if the activity state changes from α to β at time t; and in which proj (u^{k-1},k) is the projection of u^{k-1} into the space of u^k. The minimization in (53) is performed at every t subject to (47) and (52). If such a J function could be found to satisfy this nonlinear partial differential equation, the optimal control u^{k-1} could be determined from the indicated minimization.

If J were known, determining u^k would reduce to solving

$$\left.\begin{array}{l} \min \frac{\partial J}{\partial x} u^k \\ \text{subject to (47) and (52).} \end{array}\right\} \tag{54}$$

If $m(u^k)$ is a linear function, this is a linear programming problem.

Akella and Kumar (1986), Bielecki and Kumar (1987), and Sharifnia (1988) have obtained analytic solutions for versions of this problem in which x^k and u^k are scalars. In no other cases are exact solutions to this problem knowr. Numerical solutions are equally unavailable because of the "curse of dimensionality." To overcome this difficulty, Akella et al. (1984) show that a quadratic approximation of J can produce excellent performance.

Kimemia and Gershwin ran several simulations to test a simple hierarchical policy: solve (54) at every time instant to determine u^k, and then load parts (in a manner somewhat more complex and less effective than the staircase strategy of Section 5) so that the rate of loading parts was close to u^k. This worked well until the solution of (54) changed abruptly. (This is an important possibility since (54) is a linear program.) Very often, it changed abruptly again at the next time instant, and this led to reduced performance.

Gershwin et al. (1985) avoided this chattering by observing a behavior similar to that of a closely related problem of Rishel (1975). The continuous part of the state, x^k, is restricted to reduced dimensional surfaces whenever u^k would otherwise chatter. In the present problem, chattering is avoided by adding linear equality constraints to (54) whenever x^k reaches certain planes.

This step has the additional benefit of reducing computational effort. It is no longer necessary to solve (54) at every time instant. Instead, a series of computations is performed at every time t_c when there is a change in α^k. At those instants (54) is solved, and then solved repeatedly with additional constraints, as described above. The outcome of these calculations is a piecewise constant function of t, $u^k_{ij}(t;\alpha^k(t_c^+),\sigma^k(t_c^+))$, defined for $t>t_c$. This function is the set of target rates for level k+1. When α^k changes, the function is recalculated.

There are two kinds of states (α^k,σ^k): feasible and infeasible. *Feasible* states are those for which proj $(u^{k-1},k) \in \Omega^k(\alpha^k(t),\sigma^k(t))$. All other states are infeasible. If (α^k,σ^k) is feasible and constant for a long enough period, the strategy drives x^k to the value that minimizes $J(x^k,\alpha^k,\sigma^k,t)$. In steady state, this is a constant $z^k(\alpha^k,\sigma^k)$ which we call the *hedging point*. We have assumed that T is large enough so that the system can be assumed to be in steady state.

The hedging point represents a safety level of the surplus. Infeasible states are certain to occur eventually. While (α^k,σ^k) is infeasible, x^k must decrease, and possibly become negative. The hedging point represents a compromise between a cost for positive x^k and a cost for negative x^k. When the activities considered are production operations on parts, for example, the tradeoff is between production that is ahead of demand (and therefore can lead to inventory) and production that is behind demand (and therefore leads to starved downstream resources or unhappy customers). The hedging point need not be positive. Bielecki and Kumar show that it can sometimes be 0.

Levels in which configuration states change differ from those in which only failure states change in one important respect. In the latter, there is always some α^{k*} such that

$$\text{proj } (u^{k-1},k) \in \Omega^k(\alpha^{k*},\sigma^k). \tag{55}$$

For example, α^{k*} may be the state in which all machines are operational. In that case, all sets $\Omega^k(\alpha^k(t),\sigma^k)$ are subsets of $\Omega^k(\alpha^{k*},\sigma^k)$. In addition, proj $(\Omega^{k-1}(\alpha^{k-1},\sigma^{k-1}),k)$ is an average of $\Omega^k(\alpha^k(t),\sigma^k)$. Therefore,

$$u^{k-1} \epsilon\ \Omega^{k-1}(\alpha^{k-1},\sigma^{k-1}) \qquad (56)$$

implies (55). Equation (56) is true; it is (47) at level k-1.

On the other hand, in levels in which σ changes, there may not be a state whose capacity set contains all the others. Consequently, proj (u^{k-1},k) may not be a member of any capacity set. In that case, if (α^k, σ^k) stays constant for a long time, x^k will not approach a hedging point.

7. SIMPLE EXAMPLE

In this section, we illustrate the ideas developed in this paper with a two-part, two-machine system. There are only two phenomena in this system: failures and operations. The former are much less frequent, but of much greater duration, than the latter. Therefore, the hierarchy has three levels (including the static level). This is an example of the methods of Kimemia and Gershwin (1983) and Gershwin, Akella, and Choong (1985). An extension of this system, in which setup plays a role, is described in Gershwin (1987b), and Gershwin, Caramanis, and Murray (1988).

7.1 Description of System

Figure 7.1 illustrates the two-machine system. In this system, Machine 1 is perfectly flexible. That is, it can do operations on either part type, without time lost for changeover. It is unreliable, however: it fails at random times and stays down for random lengths of time. Machine 2 is perfectly reliable, but totally inflexible. It can only make Type 1 parts. Thus Machine 1 is shared among the two part types and Machine 2 is devoted entirely to Type 1.

The data that are specified are the demand rates for the parts (d_1 and d_2), the failure (p) and repair (r) rates, and the durations of the operations (τ_{11}, τ_{12}, and τ_{21}, where τ_{ij} is the duration of an operation on a Type j part at Machine i). We assume that at level 1, the demand rate for Type 1 parts is broken down by the machine at which the operation is performed, so that the specified demand rates are u^1_{11}, $u^1_{12}=d_2$, and u^1_{21}. Note that $d_1 = u^1_{11} + u^1_{21}$.

For this problem, Assumption 1 becomes:

τ_{11}, τ_{12}, τ_{21}, $1/u^1_{11}$, $1/d_2$, $1/u^1_{21}$ are the same order of magnitude. Then $f_3 \approx 1/\tau_{11}$.

These quantities are all much smaller than 1/r, 1/p, which are the same order of magnitude. Then $f_2 \approx r$.

$f_1 = 0$.

7.2 Level 2: Hedging point strategy

The states of the system are α, the repair state of Machine 1, an exogenous random variable; and x^2_{11}, x^2_{12}, and x^2_{21}, the surpluses. The control variables are u^2_{ij}, the level 1 flow rate of Type j parts to Machine i (ij = 11, 12, 21).

Here, (45) becomes

$$\dot{x}^2_{ij} = u^2_{ij} - u^1_{ij} \text{ for } ij = 11, 12, \text{ and } 21. \qquad (57)$$

The linear programming problem of Section 6.5, which determines u^2_{ij}, becomes

$$\min_{u^2_{ij}} \sum c_{ij}\,(x^2,\alpha)\;u^2_{ij} \tag{58}$$

subject to:

$$\tau_{11}u^2_{11} + \tau_{12}u^2_{12} \le \alpha \tag{59}$$
$$\tau_{21}u^2_{21} \le 1 \tag{60}$$
$$u^2_{ij} \ge 0 \tag{61}$$

where for ij = 11, 12, and 21 and for mn = 11, 12, and 21,

$$c_{ij}(x^2,\alpha) = \sum_{mn} A_{ijmn}(\alpha)x^2_{mn} + b_{ij}(\alpha). \tag{62}$$

Here, $c(x^2,\alpha)$ is the approximation of $\partial J/\partial x$. Satisfactory results have been obtained with diagonal A matrices, so we choose $A_{ijmn} = 0$ if $(mn) \neq (ij)$. The hedging point is then

$$z_{ij}(\alpha) = -\frac{b_{ij}(\alpha)}{A_{ijij}(\alpha)}.$$

The outcome of this calculation is a piecewise constant function of time $u^2_{ij}(t)$, as described by Gershwin et al. (1985). This function is used in the staircase strategy below, until the repair state α changes. When that happens, a new function is calculated at this level.

7.3 Level 3: Staircase strategy

Loading a Type j part into Machine 1 is eligible if:

1. The number of Type j parts made on Machine 1 is less than

$$\int_0^t u^2_{1j}(s)ds, \text{ and} \tag{63}$$

2. Machine 1 is now idle.

Loading a Type 1 part into Machine 2 is eligible if:

1. The number of Type 1 parts made on Machine 2 is less than

$$\int_0^t u^2_{21}(s)ds, \text{ and} \tag{64}$$

2. Machine 2 is now idle.

7.4 Simulation results

Figure 7.2 demonstrates how the cumulative output follows the cumulative requirements when the system is run with this strategy.

8. CONCLUSIONS AND FUTURE RESEARCH

8.1 Summary and conclusions

A hierarchical scheduling and planning strategy has been described for manufacturing system. It is based on two major propositions:

1. Capacity. No resource can function more than 100% of the time.

2. Frequency separation. We assume that the spectrum of events is discrete. The frequencies of important events are grouped into distinct clusters.

These are reasonable assumptions for a wide variety of systems. Simulation results (Akella, Choong, and Gershwin, 1984; Gershwin, 1987b; Gershwin, Caramanis, and Murray, 1988; Xie, 1988) indicates that, at least for simple systems, the hierarchy works well.

8.2 Future research

This work is in its early stages. Much work is required, both to establish the validity of the basic ideas, and to extend the class of systems to which it can be applied.

8.2.1 Technical issues

Among the important outstanding research problems are proving the conjecture that hierarchical decomposition is asymptotically optimal as times scales separate; determining how to deal with systems in which time scales are not widely separated; formulating and solving the hedging point problem with non-Markov events (such as those generated by a staircase strategy); and developing sufficiency conditions for capacity.

To improve on the staircase policy, new formulations of deterministic scheduling problems are required in which the objective is to load material as close as possible to a given rate. One such improvement is the work of Perkins and Kumar (1988), in which setup times are determined as a function of real or virtual buffer levels. Related ideas have be considered by Gershwin, Caramanis, and Murray (1988).

A research area of interest is in improvements to numerical solutions of hedging point problems. Because of the usual "curse of dimensionality," efficient numerical evaluations of the J and u functions do not exist for problems of practical size. Quadratic approximations to J have worked well for some problems, but there are no systematic, complete methods for choosing coefficients. In addition, no bounds on the quality of quadratic approximations have been determined.

8.2.2 Statistics in the hierarchy

We have not discussed at all the collection and processing of data in the hierarchy. Such problems are important because the rates of uncontrollable events are not known, as we have assumed here, but must be deduced from experience. This will require the solution of statistics problems.

Essentially, equations (22) and (26) are used in the opposite way from the way we have used them here. In the control hierarchy, the level k-1 rates of controllable activities are specified, and the corresponding level k rates or activities are determined. In the statistical hierarchy, the level k uncontrollable activities or rates are observed, and the corresponding level k-1 rates are calculated.

8.2.3 Spatial decomposition

This issue is related to the long time that parts spend in some kinds of manufacturing, particularly semiconductor fabrication. As pointed out in Section 6.3.2, conservation of flow holds at level k only if the time that parts spend in the system is much less than $1/f_k$. This suggests a more general structure of hierarchies for large systems: let k be the highest level such that the time that parts spend in the system is much less than $1/f_k$. Then at level k and above, the hierarchy is exactly as described here.

At level k+1, the system is divided into subsystems (level k+1 cells). In one or more of the cells, there is an event that occurs with frequency near f_{k+1} (in the sense of Assumption 1), and the time that parts spend in each of the cells is much less than $1/f_{k+1}$. Each cell is further divided as required. In this way, a pyramid-shaped hierarchy is constructed.

Preliminary work in extending the Kimemia-Gershwin formulation to systems with both operation and queuing delay and is described in Lou et al. (1987) and Van Ryzin (1987).

8.2.4 Part type and resource aggregation

Some extensions include the reduction of the problem size at higher levels. This requires aggregation of activities (so that one considers, for example, large classes of part types, rather than individual types) and of resources (so that the smallest unit can be a cell or workshop or even factory, rather than a machine).

8.2.5 Multiple routes

In this paper, we have assumed that the routes of material through the system are specified in advance. This fails to take full advantage of the flexibility of the system. For the failure-only hierarchy, Maimon and Gershwin (1988) extend the work of Kimemia and Gershwin (1983) to allow multiple routes, and this is introduced into the multi-level hierarchy by Gershwin, Caramanis, and Murray (1988).

ACKNOWLEDGMENTS

This work was supported by the Defense Advanced Research Projects Agency and monitored by ONR under contract N00014-85-K-0213. I am grateful for the helpful comments of Professor Y. C. Ho, Professor Michael Caramanis, Mr. Gerry Feigin and Mr. Feng Lin and for the simulation work of Mr Peter Murray.

REFERENCES

P. Afentakis, B. Gavish, U. Karmarkar (1984), "Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multi-stage Assembly Systems," *Management Science*, Vol. 30, No. 2, February 1984, pp. 222-239.

R. Akella, Y. F. Choong, and S. B. Gershwin (1984), "Performance of Hierarchical Production Scheduling Policy," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, Vol. CHMT-7, No. 3, September, 1984.

R. Akella and P. R. Kumar (1986), "Optimal Control of Production Rate in a Failure Prone Manufacturing System, *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 2, pp. 116-126, February, 1986.

T. Bielecki, and P. R. Kumar (1986), "Optimality of Zero-Inventory Policies for Unreliable Manufacturing Systems", Coordinated Science Laboratory, University of Illinois Working Paper; *Operations Research*, to appear.

C. A. Caromicoli (1987), "Time Scale Analysis Techniques for Flexible Manufacturing Systems," MIT EECS Master of Science Thesis, MIT Laboratory for Information and Decision Systems Report LIDS-TH-1725, December, 1987.

C. A. Caromicoli, A. S. Willsky, and S. B. Gershwin (1987), "Multiple Time Scale Analysis Techniques of Manufacturing Systems," MIT Laboratory for Information and Decision Systems Report LIDS-TH-1727, December, 1987.

G. R. Bitran, E. A. Haas, and A. C. Hax (1981), "Hierarchical Production Planning: A Single-Stage System," *Operations Research*, Vol. 29, No. 4, July-August, 1981, pp. 717-743.

Y. F. Choong (1988), "Flow Control Approach for Batch Production Scheduling with Random Demand," MIT Ph.D. Thesis, MIT Laboratory for Manufacturing and Productivity Report LMP-88-010, May 1988.

F. Delebecque, J. P. Quadrat, and P. V. Kokotovic (1984), "A Unified View of Aggregation and Coherency in Networks and Markov Chains," *International Journal of Control*, Vol. 40, No. 5, November, 1984.

M. A. H. Dempster, M. L. Fisher, L. Jansen, B. J. Lageweg, J. K. Lenstra, and A. H. G. and Rinnooy Kan, (1981), "Analytical Evaluation of Hierarchical Planning Systems," *Operations Research*, Vol. 29, No. 4, July-August, 1981, pp. 707-716.

S. B. Gershwin (1986), "Stochastic Scheduling and Setups in a Flexible Manufacturing System," in *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*, Ann Arbor, Michigan, August, 1986, pp. 431-442.

S. B. Gershwin (1987a), "A Hierarchical Framework for Discrete Event Scheduling in Manufacturing Systems," In *Discrete Event Systems: Models and Applications, IIASA* Conference, *Sopron, Hungary, August 3-7, 1987*, Edited by P. Varaiya and A. B. Kurzhanski, Number 103 of the series *Lecture Notes in Control and Information*

Sciences, Springer-Verlag.

S. B. Gershwin (1987b), "A Hierarchical Framework for Manufacturing System Scheduling: A Two-Machine Example," *Proceedings of the 26th IEEE Conference on Decision and Control*, Los Angeles, California, December, 1987

S. B. Gershwin, M. Caramanis, and P. Murray (1988), "Simulation Experience with a Hierarchical Scheduling Policy for a Simple Manufacturing System," *Proceedings of the 27th* IEEE Conference on Decision and Control, Austin, Texas, December, 1988.

S. B. Gershwin, R. Akella, and Y. F. Choong (1985), "Short-Term Production Scheduling of an Automated Manufacturing Facility," *IBM Journal of Research and Development*, Vol. 29, No. 4, pp 392-400, July, 1985.

S. C. Graves (1981), "A Review of Production Scheduling," *Operations Research*, Vol. 29, No. 4, July-August, 1981, pp. 646-675.

S. C. Graves (1982), "Using Lagrangean Relaxation Techniques to Solve Hierarchical Production Planning Problems," *Management Science*, Vol. 28, No. 3, March 1982, pp. 260-275.

A. C. Hax and H. C. Meal (1975), "Hierarchical Integration of Production Planning and Scheduling," North Holland/TIMS, Studies in Management Sciences, Vol. 1, *Logistics*.

J. Kimemia and S. B. Gershwin (1983), "An Algorithm for the Computer Control of a Flexible Manufacturing System," *IIE Transactions* Vol. 15, No. 4, pp 353-362, December, 1983.

B. J. Lageweg, J. K. Lenstra, and A. H. G. and Rinnooy Kan (1977), "Job-Shop Scheduling by Implicit Enumeration," *Management Science*, Vol. 24, No. 4, December 1977, pp. 441-450.

B. J. Lageweg, J. K. Lenstra, and A. H. G. and Rinnooy Kan (1978), "A General Bounding Scheme for the Permutation Flow-Shop Problem," *Operations Research*, Vol. 26, No. 1, January-February 1978, pp. 53-67.

C. M. Libosvar (1988a), "Hierarchies in Production Management and Control: A Survey," MIT Laboratory for Information and Decision Systems Report LIDS-P-1734, January, 1988.

C. M. Libosvar (1988b), "Hierarchical Production Management The Flow-Control Layer," Thesis for the degree of Docteur de l'Universite de Metz, April, 1988.

X.-C. Lou, J. G. Van Ryzin and S. B. Gershwin (1987), "Scheduling Job Shops with Delays," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, North Carolina, March-April 1987.

O. Z. Maimon and S. B. Gershwin (1987), "Dynamic Scheduling and Routing For Flexible Manufacturing Systems that have Unreliable Machines," *Operations Research*, to appear.

O. Z. Maimon and G. Tadmor (1986), "Efficient Low-Level Control of Flexible Manufacturing Systems," MIT Laboratory for Information and Decision Systems Report LIDS-P-1571, June, 1986.

E. F. P. Newson (1975a), "Multi-Item Lot Size Scheduling by Heuristic, Part I: With Fixed Resources," *Management Science*, Vol. 21, No. 10, June 1975, pp. 1186-1193.

E. F. P. Newson (1975b), "Multi-Item Lot Size Scheduling by Heuristic, Part I: With Variable Resources," *Management Science*, Vol. 21, No. 10, June 1975, pp. 1194-1203.

C. H. Papadimitriou and P. C. Kannelakis (1980), "Flowshop Scheduling with Limited Temporary Storage," *Journal of the ACM*, Vol. 27, No. 3, July, 1980.

J. Perkins and P. R. Kumar (1988), "Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems," *Proceedings of the 27th IEEE Conference on Decision and Control*, Austin, Texas, December, 1988.

P. J. Ramadge and W. M. Wonham (1985), "Supervisory Control of a Class of Discrete Event Processes," Systems Control Group Report No. 8515, University of Toronto.

Rishel, R. "Dynamic Programming and Minimum Principles for Systems with Jump Markov Disturbances", *SIAM Journal on Control*, Vol.13, No.2 (February 1975).

V. R. Saksena, J. O'Reilly, and P. V. Kokotovic (1984), "Singular Perturbations and Time-Scale Methods in Control Theory: Survey 1976-1983", *Automatica*, Vol. 20, No. 3, May, 1984.

A. Sharifnia (1988), "Production Control of a Manufacturing System with Multiple Machine States," *IEEE Transactions on Automatic Control*, Vol. AC-33, No. 7, pp. 620-625, July, 1988.

H. M. Wagner and T. M. Whitin (1958), "Dynamic Version of the Economic Lot Size Model," *Management Science*, Vol. 5, No. 1, October, 1958, pp. 89-96.

J. G. Van Ryzin (1987), "Control of Manufacturing Systems with Delay," MIT EECS Master of Science Thesis, MIT Laboratory for Information and Decision Systems Report LIDS-TH-1676, June, 1987.

X.-L. Xie (1988) "Hierarchical Production Control of a Flexible Manufacturing System," SAGEP Project, INRIA-Lorraine, Vandoeuvre, France.

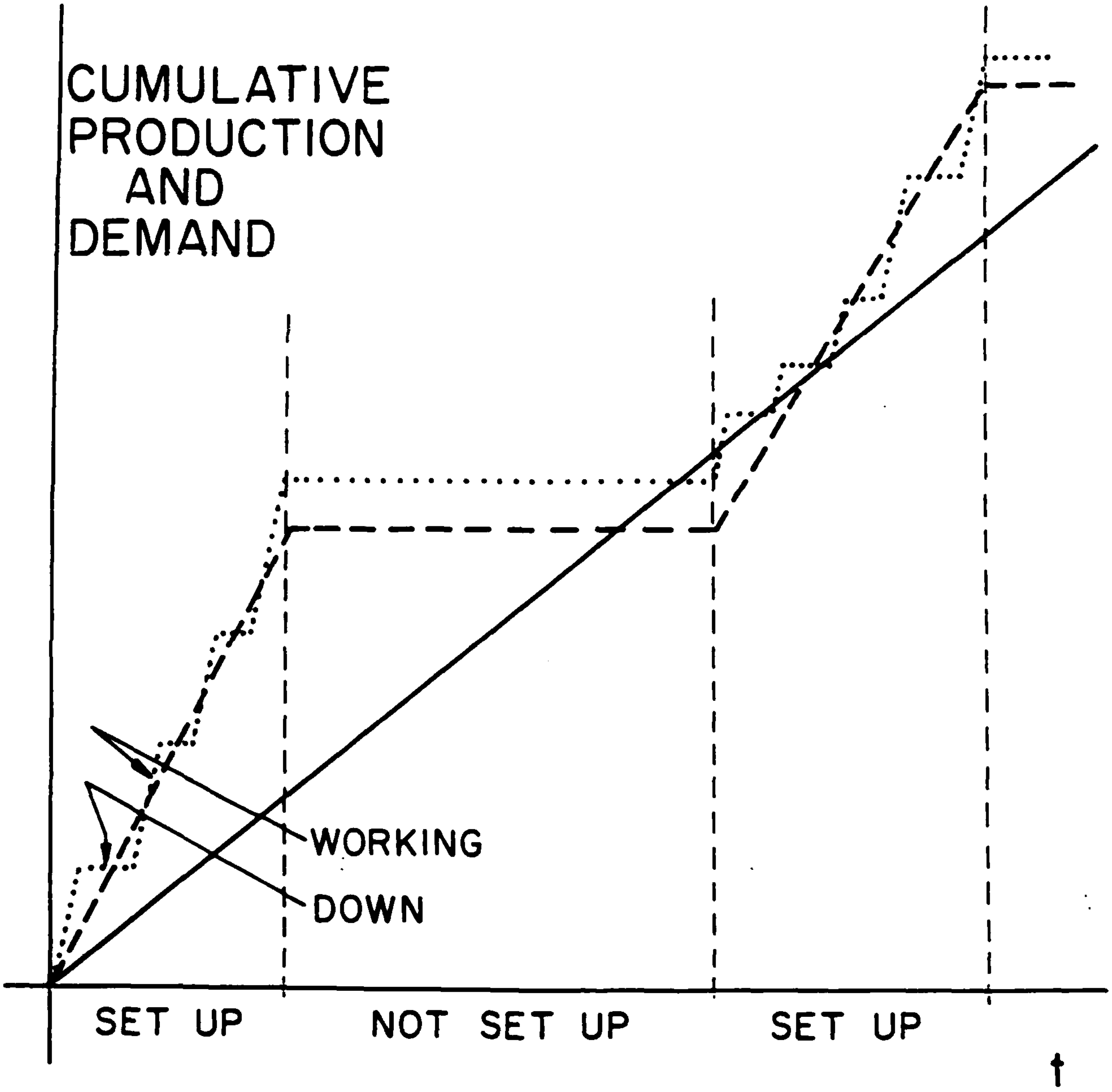
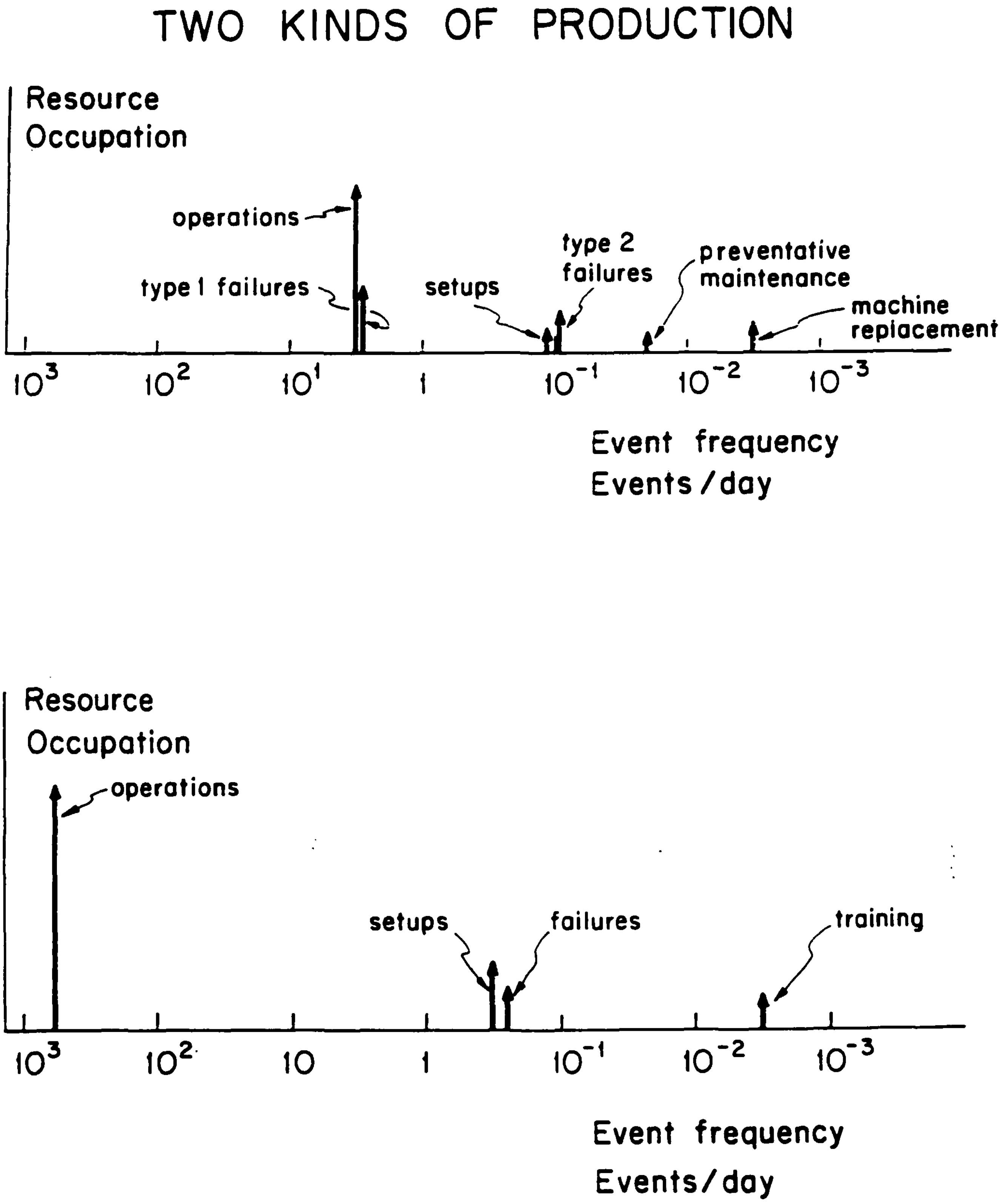
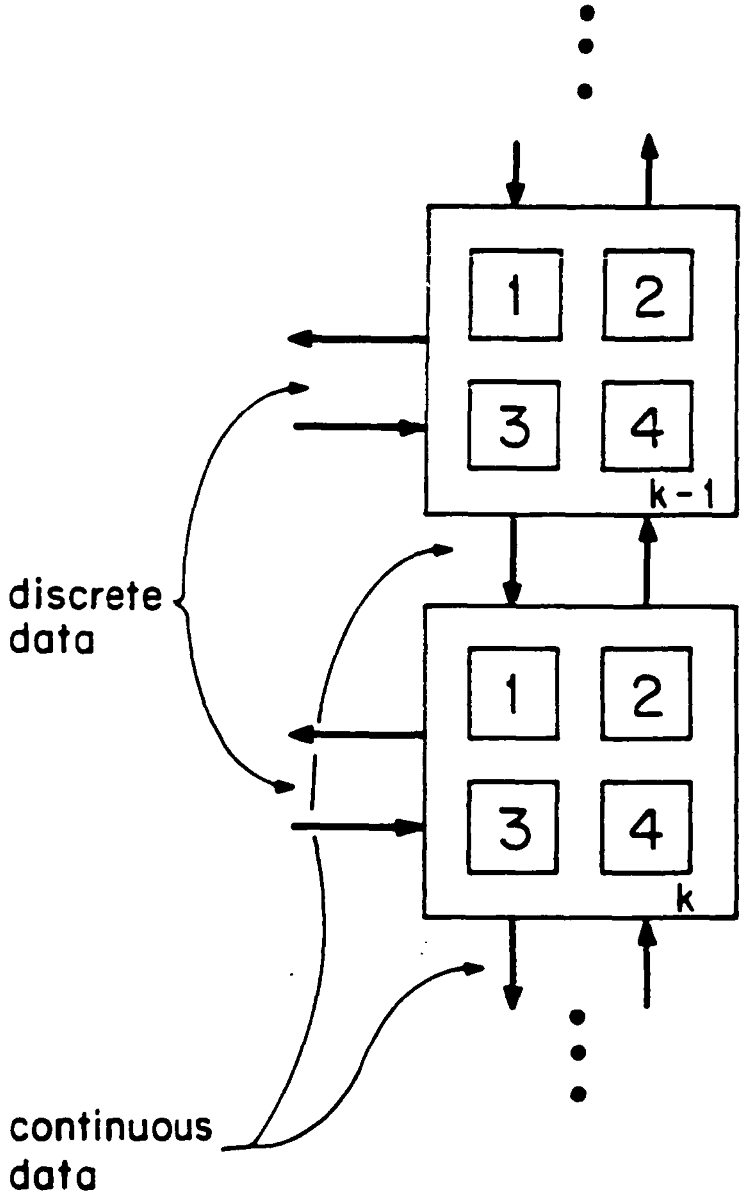


Figure 1.1 Production and Other Events

Figure 2.1 Two Spectra



ARCHITECTURE OF THE HIERARCHY



Control Problems

1- Calculate u^k from u^{k-1} and α^k.

2- Calculate α^k from u^{k-1}.

Statistics Problems

3- Calculate u^k from u^{k+1}.

4- Calculate u^k from α^{k+1}.

Figure 4.1 Hierarchy

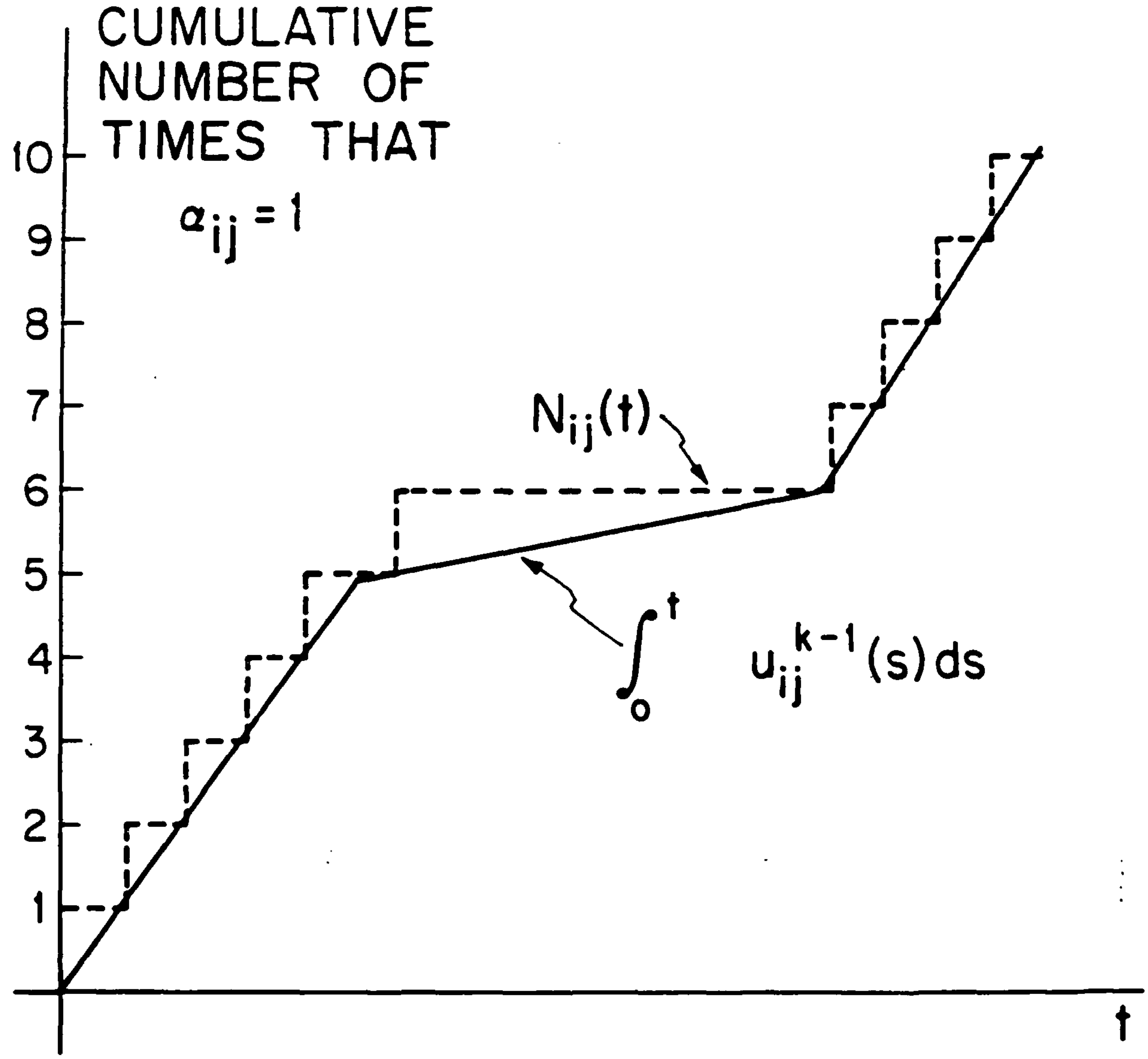


Figure 5.1 Staircase Strategy

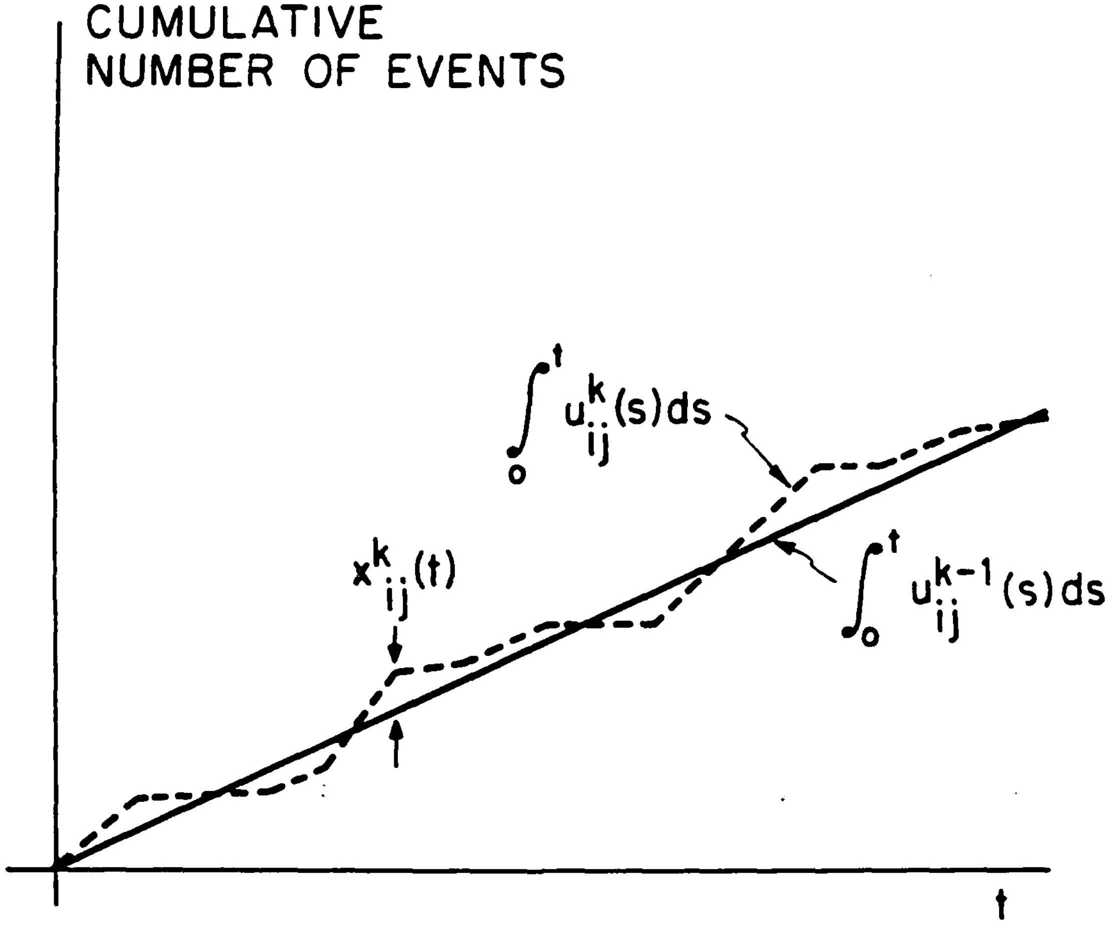


Figure 6.1 Surplus

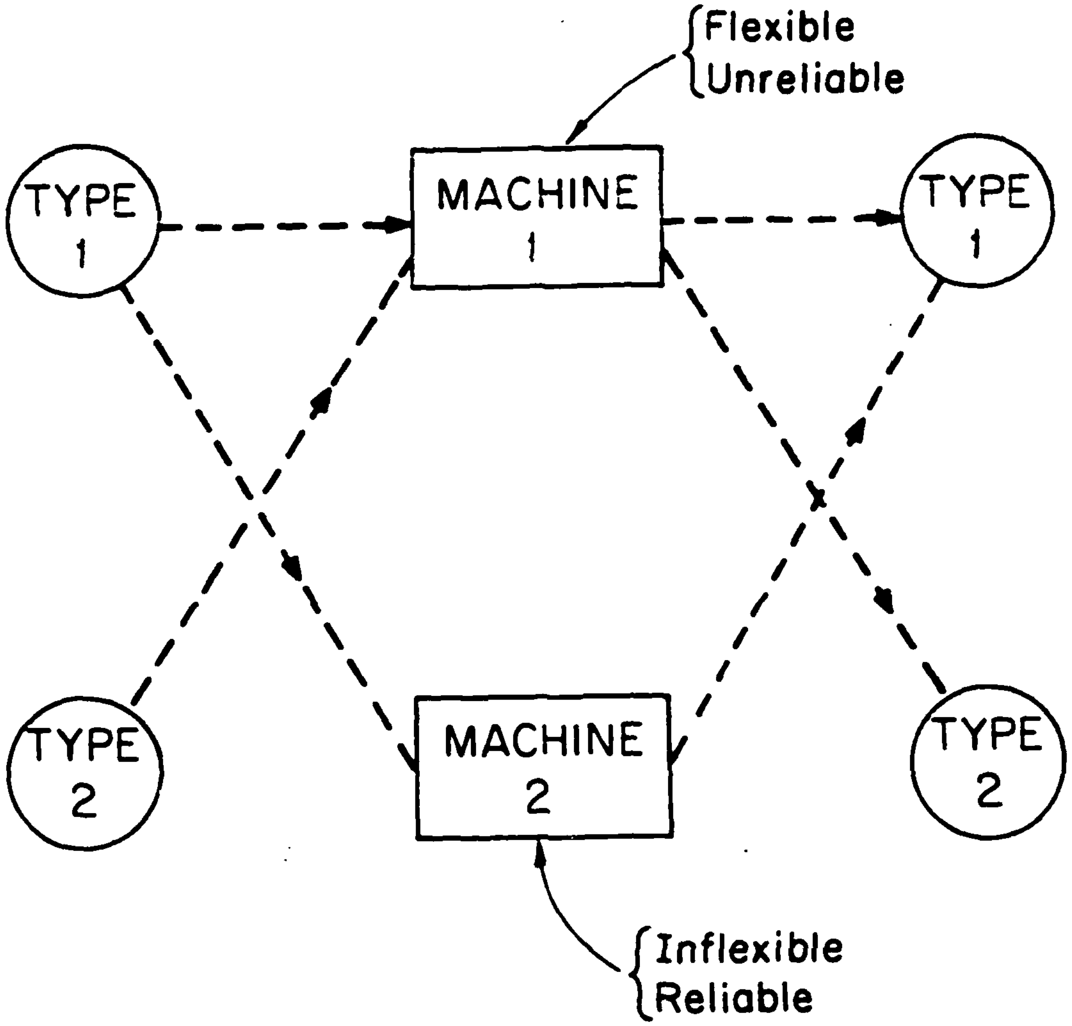


Figure 7.1 Simple System

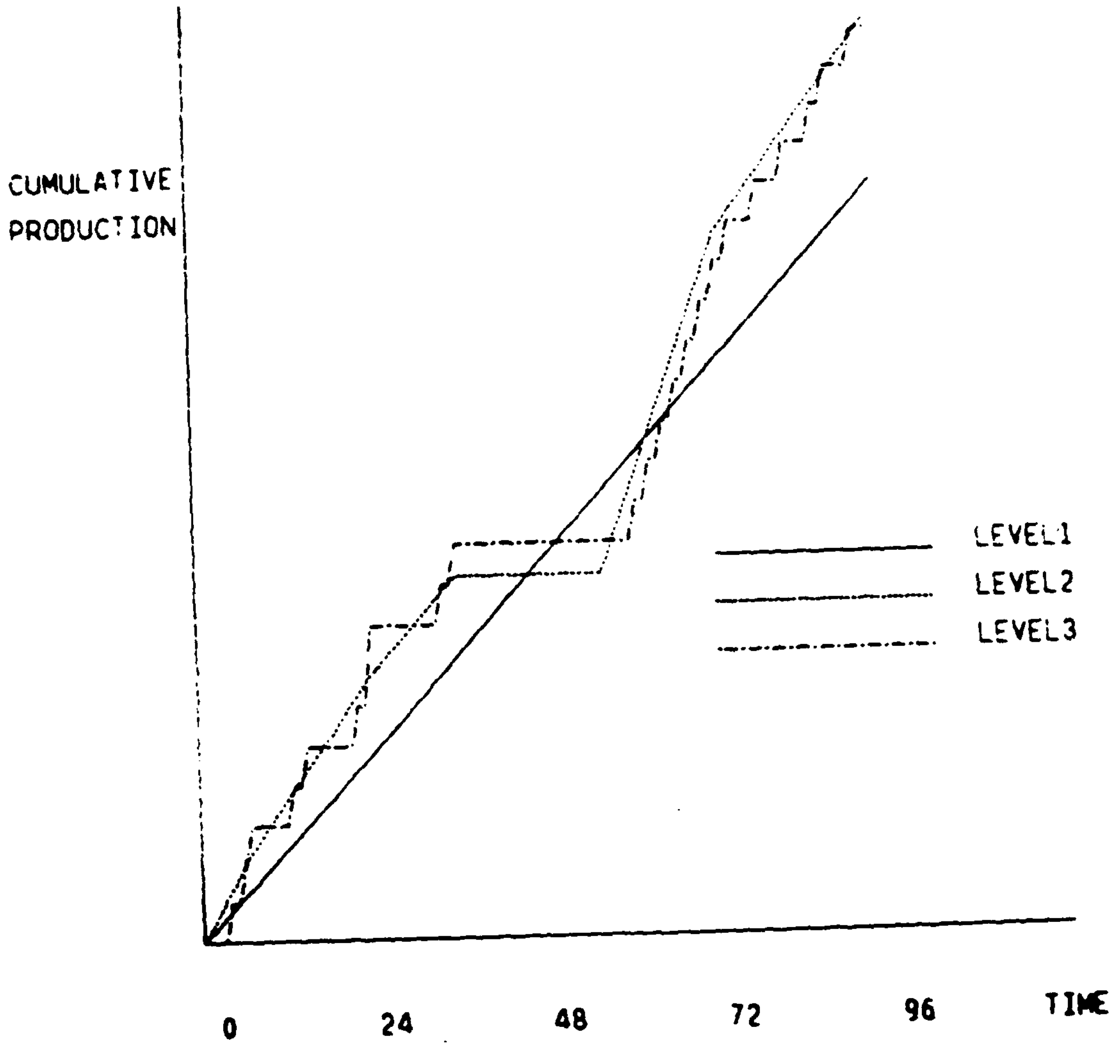


Figure 7.2 Behavior of Strategy